



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,767	12/08/2003	Richard P. Himmer	120 04983US	3406
128 7590 07/17/2009 HONEYWELL INTERNATIONAL INC. PATENT SERVICES 101 COLUMBIA ROAD P O BOX 2245 MORRISTOWN, NJ 07962-2245				
EXAMINER				
WANG, BEN C				
ART UNIT		PAPER NUMBER		
2192				
MAIL DATE		DELIVERY MODE		
07/17/2009		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/729,767

**Applicant(s)**

HIMMER ET AL.

**Examiner**

BEN C. WANG

**Art Unit**

2192

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 05 May 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1, 2, 4-8, 16 and 17 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1, 2, 4-8, 16, and 17 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SI/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. Applicant's amendment dated May 5, 2009, responding to the Office action mailed February 5, 2009 provided in the rejection of claims 1-2, 4-8, 16 and 17 wherein claims 1, 2, 5, 6, and 17 have been amended.

Claims 1-2, 4-8, 16 and 17 remain pending in the application and which have been fully considered by the examiner.

The independent claim grouping was inadvertently grouped as "claim 1, 6 and 17" instead of -- claim 1, 5 and 17 -- (stated on page 5, fourth paragraph)

The independent/associated claim groupings were inadvertently grouped as "independent claim 1, 6 and 17" and "dependent claim 2, 4, 5, 7 and 8" instead of -- claims 1, 5 and 17 -- and -- claims 2, 4, 6, 7 and 8 -- respectively (stated on page 6, first non-full paragraph)

Applicant's arguments with respect to claims currently amended have been fully considered but are not persuasive. Please see the section of "Response to Arguments" for details.

2. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened

Art Unit: 2192

statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

***Claim Rejections – 35 USC § 103(a)***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-2, 4-8 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hammack et al. (Pat. No. US 6,449,624 B1) (hereinafter 'Hammack' ) in view of Dardinski et al. (WO 00/70417) (hereinafter 'Dardinski')
4. **As to claim 1** (Currently Amended), Hammack discloses a source control system for a process control system (e.g., Fig. 1 – a process control system; Col. 3, Lines 49-52 – a process control system includes a process controller connected to one or more host workstations or computers via a communication network such as an Ethernet connection or the like), comprising:
  - a processor in a process control system (e.g., Col. 2, Lines 36-39 – the inventive system further includes a configuration routine and a version

control routine, both of which are stored in the computer-readable medium and configured to be executed by the processor);

- a database accessible by said processor to store information associated with an object under source control to be checked-out (e.g., Col. 12, Lines 66 through Col. 13, Lines 7 – the VCAT [Version Control and Audit Trail system] system may further include functionality that permits the user to request a status update for all of the items in the configuration database to ensure that those items that are checked out are indicated as such via checkmark or the like); and
- a check-out function operable on said processor to check-out said object (e.g., Fig. 6 – element of 120 – “Check Out” , “CheckOut Recursive”); to use said information to determine whether any dependent objects exist and whether at least one parent object exists, and to automatically check-out said existing dependent objects, wherein said stored information includes a reference to said existing parent object (e.g., Col. 11, Lines 24-29 – the VCAT system preferably determines during each check-out operation which other versionable items need to be checked out in order to modify the configuration of an item; the modification of these other versionable items may be referred to as “consequential changes.”, 45-48 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable;

Art Unit: 2192

Figure 4, elements 98 – VCAT System; 100 – Configuration Database; column 11, lines 45-57 - Because the configuration of the process 10 (*said information*) is set forth in a hierarchical manner, the VCAT system 98 must allow for checking out items have subordinate items (*said existing dependent objects*) that are also versionable. Accordingly, during a check-out operation, the user is provided with the option of recursively checking out any such items ...; figure 6, element 120 - CheckOut Recursive – emphasis added) wherein said stored information includes a reference to said existing parent object (e.g., figure 12, elements 208 – Display Parent Button, 210 – Drill Down Button; column 23, lines 36-56 - .. The Drill Down button 208 provides the user with the option to display (to the extent possible) the configuration information for a subordinate items of the originally selected item in a graphical format ... The Display Parent button 208 provides the use with the option to return to a graphical view of the parent item (*said existing parent object*) by removing the view of the subordinate item ...)

Further, Hammack discloses the data stored in an XML document is accessed in accordance with an object model that provides for parsing the document to create a data tree structure having a plurality of nodes associated with the version control data (e.g., Col. 22, Lines 52-58), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Process Control Configuration System with Parameterized Objects*, Dardinski discloses wherein said object being checked

Art Unit: 2192

out is a user defined template that is derived from a preconfigured object (e.g., Sec. 1.2.2 Managing Object Types, 1<sup>st</sup> Para - ... the user picks an already existing object type in the type hierarchy to act as its 'template' type, or object type to be used to create from ...; 2<sup>nd</sup> bullet - ... the new object type is created by using an existing object type as its parent ...), and wherein said existing dependent objects are children user defined templates of said object being checked out or instances of said object being checked out or of said children user defined templates (e.g., P. 8, 1<sup>st</sup> Para - ... icons or other appearance information identified in a 'parent configuration object is passed on to its children. Inherited information may be overridden ...; P. 26, 1<sup>st</sup> Para - ...The derived Parameterized Object gets its parameters by inheriting them from the defining object and by adding its own through local Parameter Definition associations; Sec. 1.3 Parameterized Object Connections, 1<sup>st</sup> Para - An IDA configuration consists not only of objects, but objects which are related to each other in a number of ways ... These relationships are all called connections; P. 47, the last bullet - Each instance of a Connection Endpoint has a reference to a Point Placeholder; P. 48, Summarizing relationships, all bullets)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dardinski into the Hammack's system to further provide other limitations stated above in the Hammack system.

The motivation is that it would further enhance the Hammack's system by taking, advancing and/or incorporating Dardinski's system which offers significant

Art Unit: 2192

advantages the apparatus employ object that inherit parameters from their ancestors; changes to an ancestor during configuration are effective as to its descendant objects and can be used to model and define control algorithms for process control, environmental control, industrial and other control system as once suggested by Dardinski (e.g., P. 4, Summary of the Invention, Parameterized Objects)

5. **As to claim 2** (Currently Amended) (incorporating the rejection in claim 1), Hammack discloses the system further comprising: a propagation function operable on said processor to propagate changes made to said object being checked out to said existing dependent objects and to said existing parent object, when said object being checked out is saved (e.g., Col. 29, Lines 11-16 – the configuration management system wherein the configuration routine is adapted to be executed by the processor to make changes to a first process control element and to propagate changes to other process control elements that are affected by the changes made to the first process control element; Figure 4, elements 98 – VCAT System; 100 – Configuration Database; column 11, lines 45-57 - Because the configuration of the process 10 (*said information*) is set forth in a hierarchical manner, the VCAT system 98 must allow for checking out items have subordinate items (*said existing dependent objects*) that are also versionable. Accordingly, during a check-out operation, the user is provided with the option of recursively checking out any such items ...; figure 6, element 120 - CheckOut Recursive – emphasis added) wherein said stored information



Art Unit: 2192

includes a reference to said existing parent object (e.g., figure 12, elements 208 – Display Parent Button, 210 – Drill Down Button; column 23, lines 36-56 - .. The Drill Down button 208 provides the user with the option to display (to the extent possible) the configuration information for a subordinate items of the originally selected item in a graphical format ... The Display Parent button 208 provides the use with the option to return to a graphical view of the parent item (*said existing parent object*) by removing the view of the subordinate item ...)

Dardinski discloses wherein said stored information includes a reference to a parent object (e.g., P. 9, Connection Validation and Configuration, 3<sup>rd</sup> Para – ... to establish a relationship between those two objects ... validates that relationship ...)

6. **As to claim 4** (Original) (incorporating the rejection in claim 1), Hammack discloses the system wherein said stored information is at least one selected from the group consisting of: a name, a version number, a type and a status (e.g., Col. 20, Lines 15-20 – Furthermore, it is preferred that key words and labels be utilized to identify attributes such as object type and properties; examples of property labels are "NAME" and "DESCRIPTION"; Col. 13, Lines 58-66 – data representative of each prior configuration of an item is stored in the version control database together with data reflective of a version assigned thereto; the version is preferably identified by number, but may be indicated in any other manner)

7. **As to claim 5** (Currently Amended), Hammack discloses a method of automatic check-out for a source control system in a process control system, comprising:

- storing information associated with an object (e.g., Fig. 3; Col. 6, Lines 25-43 – with reference now to Fig. 3, the data stored in the configuration database may be presented to a user via a configuration database administrative interface such as Delta V® Explorer, which will hereinafter be referred to as “the Explorer system”; the Explorer system sets forth a configuration hierarchy in a windows-type environment having a suite of configuration tools for modifying the elements of the hierarchy);
- receiving a request from a user to check-out said object (e.g., Fig. 6 – element of 120 – “Check Out” , “CheckOut Recursive”);
- determining whether any dependent objects of said object being checked out exist and whether any parent objects exist based on said information, wherein said stored information includes a reference to said existing parent objects (e.g., Figure 4, elements 98 – VCAT System; 100 – Configuration Database; column 11, lines 45-57 - Because the configuration of the process 10 (said information) is set forth in a hierarchical manner, the VCAT system 98 must allow for checking out items have subordinate items (said existing dependent objects) that are also versionable. Accordingly, during a check-out operation, the user is provided with the option of recursively checking out any such items ...;

- figure 6, element 120 - CheckOut Recursive – emphasis added) wherein said stored information includes a reference to said existing parent object (e.g., figure 12, elements 208 – Display Parent Button, 210 – Drill Down Button; column 23, lines 36-56 - .. The Drill Down button 208 provides the user with the option to display (to the extent possible) the configuration information for a subordinate items of the originally selected item in a graphical format ... The Display Parent button 208 provides the use with the option to return to a graphical view of the parent item (*said existing parent object*) by removing the view of the subordinate item ...);
- automatically checking-out said existing dependent objects when said object is checked-out; and providing a status to said user (e.g., Col. 11, Lines 24-29 – the VCAT system preferably determines during each check-out operation which other versionable items need to be checked out in order to modify the configuration of an item; the modification of these other versionable items may be referred to as “consequential changes.”, 45-48 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable).

Further, Hammack discloses the data stored in an XML document is accessed in accordance with an object model that provides for parsing the document to create a data tree structure having a plurality of nodes associated with the version control data (e.g., Col. 22, Lines 52-58), but does not explicitly disclose other limitations stated below.

Art Unit: 2192

However, in an analogous art of *Process Control Configuration System with Parameterized Objects*, Dardinski discloses wherein said object being checked out is a user defined template that is derived from a preconfigured object (e.g., Sec. 1.2.2 Managing Object Types, 1<sup>st</sup> Para - ... the user picks an already existing object type in the type hierarchy to act as its 'template' type, or object type to be used to create from ...; 2<sup>nd</sup> bullet - ... the new object type is created by using an existing object type as its parent ...), and wherein said existing dependent objects are children user defined templates of said object being checked out or instances of said object or of said children user defined templates (e.g., P. 8, 1<sup>st</sup> Para - ... icons or other appearance information identified in a 'parent configuration object is passed on to its children. Inherited information may be overridden ...; P. 26, 1<sup>st</sup> Para - ...The derived Parameterized Object gets its parameters by inheriting them from the defining object and by adding its own through local Parameter Definition associations; Sec. 1.3 Parameterized Object Connections, 1<sup>st</sup> Para - An IDA configuration consists not only of objects, but objects which are related to each other in a number of ways ... These relationships are all called connections; P. 47, the last bullet - Each instance of a Connection Endpoint has a reference to a Point Placeholder; P. 48, Summarizing relationships, all bullets)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dardinski into the Hammack's system to further provide other limitations stated above in the Hammack system.

The motivation is that it would further enhance the Hammack's system by taking, advancing and/or incorporating Dardinski's system which offers significant advantages the apparatus employ object that inherit parameters from their ancestors; changes to an ancestor during configuration are effective as to its descendant objects and can be used to model and define control algorithms for process control, environmental control, industrial and other control system as once suggested by Dardinski (e.g., P. 4, Summary of the Invention, Parameterized Objects)

8. **As to claim 6** (Currently Amended) (incorporating the rejection in claim 5), Hammack discloses the method further comprising: sorting said existing dependent objects so that said existing parents precede children (e.g., Col. 11, Lines 48-51 – in one embodiment, if a recursive check-out or check-in is selected by the user, the VCAT system generates a dialog window that provides the user with a list of versionable, subordinate items that may be checked out (or checked-in))

9. **As to claim 7** (Previously Presented) (incorporating the rejection in claim 5), Hammack discloses the method wherein one of said existing dependent objects is a derivation child of said object (e.g., Col. 11, Lines 45-55 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable)

10. **As to claim 8** (Original) (incorporating the rejection in claim 7), Hammack discloses the method further comprising: automatically checking-out a derivation child only if said derivation child is checked-in (e.g., Col. 11, Lines 45-55 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable)

11. **As to claim 17** (Currently Amended), Hammack discloses a computer readable medium having executable instructions stored thereon to perform a method of automatic check-out for a source control system in a process control system, said method comprising:

- storing information associated with an object (e.g., Fig. 3; Col. 6, Lines 25-43 – with reference now to Fig. 3, the data stored in the configuration database may be presented to a user via a configuration database administrative interface such as Delta V® Explorer, which will hereinafter be referred to as “the Explorer system”; the Explorer system sets forth a configuration hierarchy in a windows-type environment having a suite of configuration tools for modifying the elements of the hierarchy);
- receiving a request from a user to check-out said object (e.g., Fig. 6 – element of 120 – “Check Out” , “CheckOut Recursive”);
- determining whether any dependent objects of said object being checked-out exist and whether at least one parent object of said object exists

- based on said information, wherein said stored information includes a reference to said existing parent object (e.g., Figure 4, elements 98 – VCAT System; 100 – Configuration Database; column 11, lines 45-57 - Because the configuration of the process 10 (*said information*) is set forth in a hierarchical manner, the VCAT system 98 must allow for checking out items have subordinate items (*said existing dependent objects*) that are also versionable. Accordingly, during a check-out operation, the user is provided with the option of recursively checking out any such items ...; figure 6, element 120 - CheckOut Recursive – emphasis added) wherein said stored information includes a reference to said existing parent object (e.g., figure 12, elements 208 – Display Parent Button, 210 – Drill Down Button; column 23, lines 36-56 - .. The Drill Down button 208 provides the user with the option to display (to the extent possible) the configuration information for a subordinate items of the originally selected item in a graphical format ... The Display Parent button 208 provides the use with the option to return to a graphical view of the parent item (*said existing parent object*) by removing the view of the subordinate item ...);
- automatically checking-out said existing dependent objects when said object being checked-out is checked-out; and providing a status to said user (e.g., Col. 11, Lines 24-29 – the VCAT system preferably determines during each check-out operation which other versionable items need to be checked out in order to modify the configuration of an item; the modification of these other versionable items may be referred to as

"consequential changes.", 45-48 – because the configuration of the process is set forth in a hierarchal manner, the VCAT system must allow for checking out items having subordinate items that are also versionable). Further, Hammack discloses the data stored in an XML document is accessed in accordance with an object model that provides for parsing the document to create a data tree structure having a plurality of nodes associated with the version control data (e.g., Col. 22, Lines 52-58), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Process Control Configuration System with Parameterized Objects*, Dardinski discloses wherein said object being checked-out is a user defined template, and wherein said existing dependent objects are children user defined templates of said object being checked-out or instances of said object being checked-out or of said children user defined templates (e.g., P. 8, 1<sup>st</sup> Para - ... icons or other appearance information identified in a 'parent configuration object is passed on to its children. Inherited information may be overridden ...; P. 26, 1<sup>st</sup> Para - ...The derived Parameterized Object gets its parameters by inheriting them from the defining object and by adding its own through local Parameter Definition associations; Sec. 1.3 Parameterized Object Connections, 1<sup>st</sup> Para - An IDA configuration consists not only of objects, but objects which are related to each other in a number of ways ... These relationships are all called connections; P. 47, the last bullet - Each instance of a Connection Endpoint has a reference to a Point Placeholder; P. 48, Summarizing relationships, all bullets)



Art Unit: 2192

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dardinski into the Hammack's system to further provide other limitations stated above in the Hammack system.

The motivation is that it would further enhance the Hammack's system by taking, advancing and/or incorporating Dardinski's system which offers significant advantages the apparatus employ object that inherit parameters from their ancestors; changes to an ancestor during configuration are effective as to its descendant objects and can be used to model and define control algorithms for process control, environmental control, industrial and other control system as once suggested by Dardinski (e.g., P. 4, Summary of the Invention, Parameterized Objects)

***Claim Rejections – 35 USC § 102(b)***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(b) that form the basis for the rejections under this section made in this office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. Claim 16 is rejected under 35 U.S.C. 102(b) as being anticipated by Dardinski

13. **As to claim 16** (Previously Presented), Dardinski discloses a computer readable medium having executable instructions stored thereon to perform a method version control (e.g., P. 92, Sec. 1.8 Version Control), said method comprising:

when checking-in an object, determining relationships of said object by:

- determining whether said object being checked-in has a first derivation parent; adding a name and a version of said first derivation parent to a list of object relationships, if said object being checked-in has said first derivation parent (e.g., Sec. 1.8.2.2 – Check-In, 1<sup>st</sup> Para – At check-in (as depicted in Figure 47), version 2.0 of the object now “officially” exist, and both version 1.0 (*said first derivation parent*) and 2.0 (*said object*) get pointers to each other updated, effectively creating a doubly-linked list to allow traversal of version tree for this object ...);
- determining for each contained object that is contained in said object being checked-in, whether said contained object has a second derivation parent, if said object being checked-in does not have said first derivation parent; adding a name and a version of said second derivation parent to said list of object relationships, if said contained object being checked-in has said second derivation parent (e.g., page 26, last second bullet – A parameterized object may be contained with **a parameterized object collection object** (*contained object*) ... In turn, parameterized object collections may contain zero or more parameterized objects; Figure 51 – Version Control Object Model; P.

- 104, Sec. 1.8.3.6 Checked-In Parameterized Object, 4<sup>th</sup> Para – Relationship are, 3<sup>rd</sup> bullet – A checked-in version of a Parameterized Object maintains a reference to its Previous Version, in support of the requirement to be able to 'chain' down the version tree of any Parameterized object ...; Sec. 1.8.3.7 Parameterized Object (Previous Version), 1st Para - ... Each Parameterized Object is aware of its immediate ancestor (*said second derivation parent*), and its immediate descendant, in the 'geneology' hierarchy ... Relationships are: including all four bullets); and
- providing said list of object relationships (e.g., P. 8, 1<sup>st</sup> Para – The configurable objects of such an apparatus can be associated with one another in a hierarchical relationship ... such object is a descendant of another ...; 3<sup>rd</sup> Para - ... the persistent document may contain connector graphics that depict relationships between configurable objects ... Such connector graphics ... to hierarchical relationships (such as parent/child relationships))

### ***Response to Arguments***

14. Applicant's arguments filed on May 5, 2009 have been fully considered but they are not persuasive.

***In the remarks, Applicant argues that, for examples:***

(A.1) Dardinski and the combination of Dardinski and Hammack lack the follow quoted recital:

Art Unit: 2192

*"a check-out function operable on said process to check-out said object, to use said information to determine where any dependent object exist and whether at least one parent object exists, and to automatically check-out said existing dependent objects, wherein said stored information includes a reference to said existing parent object" (stated in 5<sup>th</sup> paragraph on page 5 through 1<sup>st</sup> non-full paragraph on page 6 - emphasis added)*

**(A.2)** The examiner has made no reference to any part of Dardinski to support the two determining steps and the providing step recited in independent claim 16 as quoted recitals:

*"adding a name and a version of said first derivation parent to a list of object relationships, if said object being checked-in has said first derivation parent" and "adding a name and a version of said second derivation parent to said list of object relationships, if said contained object being checked-in has said second derivation parent" (stated in page 7)*

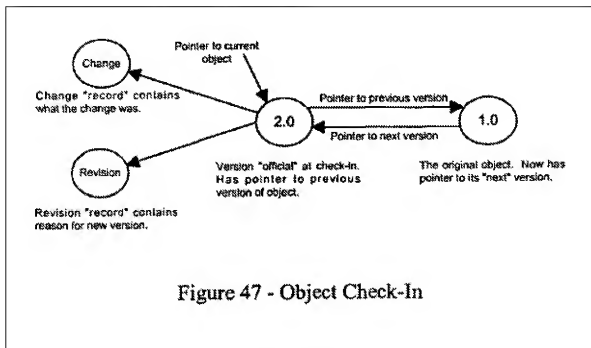
***Examiner's response:***

**(R.1)** As per the argument one (A.1), Hammack clearly teaches a check-out function operable on said process to check-out said object, to use said information to determine where any dependent object exist and whether at least one parent object exists, and to automatically check-out said existing dependent objects (e.g., figure 4, elements 98 – VCAT System; 100 – Configuration Database; column 11, lines 45-57 - Because the configuration of the process 10

Art Unit: 2192

(*said information*) is set forth in a hierarchical manner, the VCAT system 98 must allow for checking out items have subordinate items (*said existing dependent objects*) that are also versionable. Accordingly, during a check-out operation, the user is provided with the option of recursively checking out any such items ...; figure 6, element 120 - CheckOut Recursive – emphasis added) wherein said stored information includes a reference to said existing parent object (e.g., figure 12, elements 208 – Display Parent Button, 210 – Drill Down Button; column 23, lines 36-56 - .. The Drill Down button 208 provides the user with the option to display (to the extent possible) the configuration information for a subordinate items of the originally selected item in a graphical format ... The Display Parent button 208 provides the use with the option to return to a graphical view of the parent item (*said existing parent object*) by removing the view of the subordinate item ...)

**(R.2)** As per the argument two (A.2), Indeed, Dardinski discloses "*adding a name and a version of said first derivation parent to a list of object relationships, if said object being checked-in has said first derivation parent*" (e.g., Sec. 1.8.2.2 – Check-In, 1<sup>st</sup> Para – At check-in (as depicted in Exhibit A below), version 2.0 of the object now "officially" exist, and both version 1.0 (*said first derivation parent*) and 2.0 (*said object*) get pointers to each other updated, effectively creating a doubly-linked list to allow traversal of version tree for this object ...)



(Exhibit A)

Further, Dardinski teaches "adding a name and a version of said second derivation parent to said list of object relationships, if said contained object being checked-in has said second derivation parent" (e.g., page 26, last second bullet – A parameterized object may be contained with a parameterized object collection object (contained object) ... In turn, parameterized object collections may contain zero or more parameterized objects; Figure 51 – Version Control Object Model; P. 104, Sec. 1.8.3.6 Checked-In Parameterized Object, 4<sup>th</sup> Para – Relationship are, 3<sup>rd</sup> bullet – A checked-in version of a Parameterized Object maintains a reference to its Previous Version, in support of the requirement to be able to 'chain' down the version tree of any Parameterized object ...; Sec. 1.8.3.7 Parameterized Object (Previous Version), 1st Para – ... Each Parameterized Object is aware of its immediate ancestor (said second

Art Unit: 2192

*derivation parent*), and its immediate descendant, in the 'genealogy' hierarchy ...

Relationships are: including all four bullets)

### ***Conclusion***

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/

Ben C. Wang

Examiner, Art Unit 2192

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192